A Management Method of IP Multicast in Overlay Networks using OpenFlow

August 13, 2012

Yukihiro Nakagawa
Fujitsu Laboratories Ltd.
Outline

- Overlay Networks and VXLAN
- Challenges in Scalability
- Management of IP Multicast using OpenFlow
- Prototype of VXLAN Environment
- Conclusion

VXLAN: Virtual eXtensible LAN
Overlay Networks and VXLAN

- Overlay networks
  - Network virtualization is important to realize dynamic infrastructure
  - Overlay networks stretch Layer 2 networks and increase mobility of VMs

- Virtual eXtensible LAN (VXLAN)
  - VXLAN is one of overlay networking technologies being defined in IETF
  - 16 Million overlay networks can be defined for multi-tenancy
Broadcast Domain Mapping in VXLAN

- VXLAN is one of IP Multicast applications
  - Multiple broadcast domains are mapped into IP Multicast space
  - Multicast trees are dynamically configured by IGMP
  - VXLAN can be used over either Layer 2 or Layer 3

- Our use case is Layer 2 network at server edge
  - Efficient multicast replication can be achieved using IGMP snooping
  - Layer 2 switches prune the multicast trees using IGMP messages

Layer 2 switch requirements for multiple broadcast domains

<table>
<thead>
<tr>
<th>Broadcast Domain</th>
<th>L2 Switch Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional Layer 2 Network</td>
<td>VLAN</td>
</tr>
<tr>
<td>VXLAN Overlay Network</td>
<td>IP Multicast</td>
</tr>
</tbody>
</table>
Example of Multicast Path Mgmt

- IGMP Join/Leave messages to configure broadcast domains

Migration

VTEP: VXLAN Tunnel End Point

Switch’s IP Multicast capability is important
Limitation of VXLAN Scalability

■ Issue in applying VXLAN as it is to Layer 2 network
  ■ Number of BC domains is limited by number of IP Multicast groups
  ■ IGMP protocol is processed by local CPU on the switch
    • # of IGMP messages becomes bottleneck

■ Examples of 10GbE switch implementation

<table>
<thead>
<tr>
<th></th>
<th>Vender A</th>
<th>Vender B</th>
<th>Vender C</th>
<th>Vender D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch capability</td>
<td>1.28Tbps</td>
<td>1.28Tbps</td>
<td>1.28Tbps</td>
<td>1.28Tbps</td>
</tr>
<tr>
<td>VLAN table</td>
<td>4k</td>
<td>4k</td>
<td>4k</td>
<td>4k</td>
</tr>
<tr>
<td>Mac table</td>
<td>128K</td>
<td>128k</td>
<td>128K</td>
<td>120K</td>
</tr>
<tr>
<td>L3 routing table</td>
<td>16K</td>
<td>8K</td>
<td>16K</td>
<td>8k</td>
</tr>
<tr>
<td>Multicast table</td>
<td>4K</td>
<td>8K</td>
<td>8K</td>
<td>3.5K</td>
</tr>
</tbody>
</table>

We would like to make VXLAN more scalable
Proposal: IP Multicast with OpenFlow

- Configure IP Multicast path by Central controller
  - Use MAC Table and set it up proactively using OpenFlow
    → IP Multicast without using IGMP

Note: No Packet-In messages is necessary in Proactive mode
Prototype of VXLAN Environment

- We developed prototype which consists of:
  - VXLAN controller + IP Multicast Controller (Network Manager)
  - Edge switch with VTEP (VXLAN gateway)
  - OpenFlow switch

Our 10GbE switch used for Edge and OpenFlow switches.
Using OpenFlow in our Environment

- Flow table forwarding model based on OpenFlow V1.0
  - The flow table for IP multicast is mapped to the MAC address table
- Extension of output action for IP multicast
  - Output port vector is included to specify multiple destinations (see below)

```c
struct ofp_action_output_vendor {
    uint16_t type;   /* OFPAT_VENDOR. */
    uint16_t len;    /* Length is 16. */
    uint64_t portvec; /* Output port vector. */
    uint8_t pad[4];  /* Pad to 64 bits. */
};
OFP_ASSERT(sizeof(struct ofp_action_output_vendor) == 16);
```

This is vendor specific. We think multicast packet handlings should be included in OpenFlow spec for interoperability
Summary of Control Message Reduction

- Bottleneck of control plane was removed in our environment
- We compared number of messages (See Table 3 in the proceedings)
  - Examples of \# VTEP=1K and IGMPv2 are shown below

<table>
<thead>
<tr>
<th></th>
<th>Periodical messages</th>
<th>Non-periodical messages</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># of broadcast</td>
<td># of messages per second</td>
</tr>
<tr>
<td></td>
<td>domains</td>
<td>per second</td>
</tr>
<tr>
<td>IGMP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1K</td>
<td>0.1M Join</td>
<td>10 us</td>
</tr>
<tr>
<td>8K</td>
<td>0.8M Join</td>
<td>1.25 us</td>
</tr>
<tr>
<td>16M</td>
<td>1.6G Join</td>
<td>0.625 ns</td>
</tr>
<tr>
<td>Our Proposal</td>
<td>16M</td>
<td>None</td>
</tr>
</tbody>
</table>

Note: Actual \# of broadcast domains is limited by hardware resource or table size

In our application, the central controller configures IP Multicast.
We achieved more than 4K broadcast domains in our network
Conclusion

- Proposed IP Multicast with OpenFlow for Overlay Network
  - IP Multicast path is configured by the central controller
  - Use MAC Table for IP Multicast and set it up proactively using Flow Modify
  - Enhance output action to specify multiple output ports for IP Multicast

- Prototype of VXLAN Environment
  - VXLAN controller + IP Multicast Controller (Network manager)
  - Edge switch with VTEP (VXLAN gateway)
  - OpenFlow switch

- Confirmed our proposed method
  - Removed bottleneck of control plane by eliminating IGMP messages
  - Achieved more than 4K broadcast groups in our network at server edge
FUJITSU

shaping tomorrow with you